Grammatical Error Correction with (almost) no Linguistic Knowledge

Grammatical Error Correction with (almost) no Linguistic Knowledge

Roman Grundkiewicz, Marcin Junczys-Dowmunt

Adam Mickiewicz University
ul. Wieniawskiego 1, 61-712 Poznań, Poland
{romang,junczys}@amu.edu.pl

Abstract

In this work, we reinvestigate the classifier-based approach to article and preposition error correction going beyond linguistically motivated factors. We show that state-of-the-art results can be achieved without relying on a plethora of heuristic rules, complex feature engineering and advanced NLP tools. A proposed method for detecting spaces for article insertion is even more efficient than methods that use a parser. We are the first to propose and examine automatically trained word classes acquired by unsupervised learning as a substitution for commonly used part-of-speech tags. Our best models significantly outperform the top systems from CoNLL-2014 Shared Task in terms of article and preposition error correction.

1. Introduction

In the field of grammatical error correction (GEC), a large effort is made to design models and algorithms that incorporate linguistic knowledge. Heuristic rules, advanced tools or resources for natural language processing that were not created specifically with grammatical error correction in mind are commonly used. This results in a high degree of complexity with modest gains in overall performance. Results are difficult to reproduce and the integration of different systems is complicated. We believe, in accordance with Occam's Razor, that between two models that solve the same problem on similar levels of quality the simpler one is to be preferred.

In this work, we reinvestigate the classifier-based grammatical error correction paradigm by reducing its dependence on heuristic rules and advanced natural language processing tools. We focus on two of the most frequent error types among English as a second language (ESL) learners: article and preposition errors.

The only features we allow ourselves to use are simple n-gram features of: surface level tokens, part-of-speech (POS) tags, and automatically trained word classes (AWC). Where possible we try to replace POS tags with AWC tags. The latter are language-independent tags produced by clustering vector space representations of words which in turn are learnt on large unannotated text (Mikolov et al., 2013).

Our main contributions are the following: Firstly, a new contextual method for detecting omitted articles is introduced that in practice outperforms previous methods. Secondly, we are the first to apply unsupervised word classes to classifier-based GEC and GEC in general. Finally, we show that it is possible to achieve state-of-the-art results for article and preposition error correction with almost no linguistic knowledge.

The remainder of the paper is organized as follows: Section 2. reviews recent research. Data sets, classification algorithms, feature sets, and evaluation schemes are described in Section 3. Section 4. deals with detection of spaces for potential article insertions. In Section 5., we present our results and compare them with top systems from the CoNLL 2014 shared task (Ng et al., 2014). Conclusions and future work are presented in Section 6..

2. Related work

In this section we briefly discuss related work with predominantly classifier-based approaches which focused on correcting mistakes in article and preposition usage. For more comprehensive description of the field we refer the reader to the work of Leacock et al. (2010) and the recent CoNLL shared tasks (Ng et al., 2013; Ng et al., 2014).

The majority of researchers use lexical word forms, POS tags and structural information from shallow parser when designing features for article error correction classifiers. Features that encode linguistic knowledge are extracted, for example combinations of words preceding the article and a head word of the indentified noun phrase (Han et al., 2006; Gamon et al., 2008).

For instance, Rozovskaya et al. (2013) design high-level features that encode POS tags and shallow parse properties. The authors show that adding rich features to the baseline system that uses only word n-grams is helpful. However, they do not compare these rich features with simple POS n-grams.

Features used for preposition error correction are usually less complex and base on lexical forms of surrounding words (Han et al., 2010; Rozovskaya and Roth, 2010). Linguistically more complex knowledge is encoded in features that make use of various aspects of preposition complements (Tetreault and Chodorow, 2008) or additional features derived from a constituency and a dependency parse trees (Tetreault et al., 2010).

The results of Rozovskaya et al. (2013; 2014) are most similar to our work as all features are lexical, but the only type of n-grams tested in this work are pure word n-grams.

3. Experimental setting

In this work we follow the contextual classification approach to ESL grammatical error correction, which based on predefined sets of commonly confused words. The aim of a pre-trained classifier is to decide for each word that has been encountered in the text and that belongs to the confusion set, which of the possible alternatives is the most accurate in the given context.

3.1. Confusion sets

Typically, a confusion set for article and determiner error correction consists of three units: {*a, the,* ∅}[1]. This covers article insertion, deletion, and substitution errors. The distinction between *a* and *an* is usually made with heuristic rules during postprocessing. Since we made a point of not using any heuristic rules, our confusion set comprise both indefinite article variants, taking the final form: {*a, an, the,* ∅}.

In preposition error correction it is common to include in confusion set the top n most frequent English prepositions (Gamon et al., 2008; Rozovskaya and Roth, 2010; Cahill et al., 2013). We restrict ourselves to the top twelve prepositions: {*in, at, on, for, since, with, to, by, about, from, of, as*} that cover 88.6% of all preposition errors in NUCLE (see Section 3.2.). In contrast to previous studies that consider only incorrectly selected prepositions, we handle also extraneous and (for final models) missing ones.

3.2. Data sets

For training and testing our models we use various versions of two learner data resources: the NUS Corpus of Learner English and the Lang-8 corpus. A brief summary of used corpora is presented in Table 1.

Corpus	Size	$\mathbf{ER}_{art}$	$\mathbf{ER}_{prep}$
NUCLE	57,151	**6.68**	**2.34**
TS-2013	1,381	18.27	5.39
TS-2014 A0	1,312	10.23	5.08
TS-2014 A1	1,312	13.72	6.72
L8-NAIST	2,215,373	15.86	7.61
L8-WEB	3,386,887	18.55	9.22

Table 1: Basic statistics of data sets used in experiments: size in sentences and error rates (in %) for article and preposition errors.

The NUS Corpus of Learner English (NUCLE) (Dahlmeier et al., 2013) consists of 1,414 essays (57,151 sentences) which cover a wide range of topics, such as environmental pollution and health care. It was used as training data in two editions of the CoNLL shared task on Grammatical Error Correction (Ng et al., 2013; Ng et al., 2014).

We also make use of the official test sets from the shared tasks (TS-2013 and TS-2014). This data covers similar topics as NUCLE, but is smaller (1,381 and 1,312 sentences) and has higher frequencies of both error types.

By "Lang-8" we refer to a collection of posts scrapped from a language exchange social networking website named Lang-8[2]. We use the English part of the publicly available "Lang-8 Learner Corpora v1.0" (Mizumoto et al., 2012) (L8-NAIST).

Furthermore, we have scrapped recent data from the Lang-8 website which resulted in a resource (L8-WEB) that is about one and a half times larger than L8-NAIST.

[1] ∅ stands for the zero article in English.

[2] http://lang-8.com/

3.3. Classification algorithm

Our largest models are trained on over 4 million training examples represented as binary feature vectors of a length that exceeds 1.5 million features. Therefore, we decided to use the L2-regularized logistic regression from LIBLINEAR (Fan et al., 2008) which supports large-scale multi-class classification. Logistic linear regression has been used before for correction of both, article and preposition errors (Tetreault and Chodorow, 2008; Han et al., 2010; Cahill et al., 2013).

3.4. Feature sets

During experiments we use various combinations of the following features:

- *source* — a source confused word encountered in the input text, i.e. the original article or preposition.
- *tokens* — n-grams of lowercased tokens around the confused word. All n-grams have lengths between one and four, and include or are adjacent to the position of confused words.
- *POS* — n-grams of part-of-speech tags obtained by the Stanford Part-Of-Speech Tagger. The tagset consists of 43 tags.
- *AWC* — n-grams of automatic word classes created with the *word2vec* toolkit[3] (Mikolov et al., 2013). The number of clusters and vector length were set to 200. Other than that, default options were used. We learnt word vectors from 75 millions of English sentences extracted from Common Crawl data[4].
- mix_{tags} — n-grams that consist of mixed tokens and tags, e.g. for tokens w_1, w_2, w_3, and corresponding tags t_1, t_2, t_3, the mixed n-grams are: t_1w_2, w_1t_2, $t_1w_2w_3$, $w_1t_2w_3$, $w_1w_2t_3$, $t_1t_2w_3$, $w_1t_2t_3$, etc.

For each word included in a confusion set encountered in the to-be-corrected text, we extract specific features which are later converted to binary feature vectors.

It should be emphasized that experimenting with various vector space representation models, size of space dimensions and number of clusters are not within the scope of this work.

3.5. Evaluation

We use the evaluation scheme and official test sets from the CoNLL-2014 shared task (Ng et al., 2014). The system outputs submitted by participants are publicly available[5], so that we can easily compare our models with top systems from this competition. The participants were free too use all resources that were publicly available, in particular the NUCLE corpus and test set from 2013.

System performance is measured by the MaxMatch (M^2) metric (Dahlmeier and Ng, 2012) which computes an $F_{0.5}$ score for the proposed corrections against a gold standard that has been similarly annotated as NUCLE.

[3] https://code.google.com/p/word2vec/

[4] https://commoncrawl.org/

[5] http://www.comp.nus.edu.sg/~nlp/conll14st.html

The original test sets contain annotations of errors from 28 error categories. Evaluation focused on specific errors only results in very low recall in a 28 error type context, which disturbs the tuning process as well as final results due to harmonic properties of F-score. Therefore, we have modified gold standards for each test set by preserving only annotations for which the erroneous or corrected texts concern words from our confusion set (keeping deletions and insertions as well).

This method works better for us than rely on the original error categories, since many annotations that involve articles or prepositions are categorized differently (e.g. many article deletions are categorized as "local redundancy").

4. Detection of article omissions

Article omissions represent a majority of article and determiner errors, for example, in NUCLE they constitute about 61.38% of all article errors. The most common solution for detecting positions where an article might have been incorrectly omitted is to use a shallow parser to identify spaces occurring before noun phrases (Han et al., 2006; Rozovskaya et al., 2013). All noun phrases headed by a personal or demonstrative pronoun are excluded. Some research extends this by taking into account additional spaces following a preposition or a verb even when these are not identified by the parser.

On the other hand, a naive method which includes every space as a potential position for article insertion is considered to produce a lot of noise.

4.1. Detection by context comparison

We tested a new method of detection of spaces for potential article insertions based on the comparison of surrounding context. The proposed method consists of a training and a detecting stage.

During training, we extract n-grams from a text corpus consisting of l tokens to the left and r tokens to the right of each occurrence of words from the confusion set. Next, in the to-be-corrected text we flag each space for which a matching n-gram from the set of n-grams extracted during the training stage is found. Changing the minimum count c required for n-grams to be used for detection allows for control of the number of detected spaces. This procedure can be used with token n-grams and POS or AWC tags.

We estimated experimentally the values $l = 1$ and $r = 3$ for article errors, and $l = 3$, $r = 1$ for preposition errors (final models only). The n-grams were trained on a part of English Common Crawl Corpus consisting of ca. 75 million sentences.

4.2. Comparison of detection methods

We compared several methods for finding spaces for potential article insertions in the task of zero article detection. We used the entire NUCLE corpus as test set. The only positive class during evaluation (true positive) was the proper detection of a space where, according to the annotation, an article is missing. A good method should achieve a high recall and a low false positive rate (FPR). Results on NUCLE are presented in Table 2.

Method	TP	FP	FPR	R
naive	3,346	984,307	91.88	100.00
NP	2,871	**186,484**	68.20	**85.80**
$NP_{verb,prep}$	3,041	324,531	78.87	90.88
$tokens_5$	1,059	35,543	29.02	31.56
AWC_5	2,797	348,746	80.04	**83.34**
AWC_{50}	2,157	**178,290**	67.22	64.27
POS_{50}	3,167	527,534	85.85	94.37
POS_{500}	2,901	315,537	78.40	**86.44**
POS_{5000}	2,520	**170,078**	66.17	75.09

Table 2: Comparison of various methods for detecting spaces for potential article insertions. Results for NUCLE corpus: number of true positives (TP) and false negatives (FP), false-positive rate (FPR) and recall (R).

A naive method (*naive*) detects all of 984,307 spaces between words excluding spaces before and after *a, an* or *the*. A method that uses a shallow parser (*NP*) results in recall of 85.80, similarly to methods based on AWC n-grams with $c = 5$ (AWC_5) and POS n-grams (POS_{500}). But the latter almost double the number of false positives. Enforcing similar FPR requires to set $c = 50$ for AWC and $c = 5000$ for POS tags. We further evaluate these methods in the article error correction task in Section 5.1.

5. Experimental results

We use 4-fold cross validation on NUCLE to adjust threshold values of the minimum classifier confidence required to accept its prediction. During each of the steps, additional data in the form of Lang-8 corpora is added as training data. Then, we train the classifier again on the entire data and for final evaluation we use an averaged confidence threshold[6].

To prevent the classifier from keeping the input text unchanged (Cahill et al., 2013), the error rate of the training data was increased by randomly removing correct sentences. We experimentally set the error rate to 30% for article errors (keeping 873,917 and 1,660,896 sentences in L8-NAIST and L8-WEB respectively) and to 20% for preposition errors (1,219,127 sentences in L8-WEB).

The TS-2013 is used to determine an error rate for tuning data in cross validation as there is a significant disproportion in error rates[7]. We report results on both test sets, for article and for preposition models in Table 3.

5.1. Methods for detecting article omissions

In order to compare the various methods of detecting spaces for possible article insertion (Section 4.), we used the tuned L8-NAIST corpus as training data with feature set consisting of token n-grams. Results are presented in Table 3, section A.

[6]Tools and scripts that we used to perform our experiments and best performing models are made publicly available for download: `https://github.com/snukky/geccla`.

[7]The systems participating in CoNLL-2014 shared task that we compare with in Section 5.3. were free to use test data from 2013. The AMU system (Grundkiewicz and Junczys-Dowmunt, 2014) also used NUCLE for tuning with an error rate adjusted to rate observed in TS-2013.

		System description	TS-2013 P	TS-2013 R	TS-2013 $M^2_{0.5}$	TS-2014 P	TS-2014 R	TS-2014 $M^2_{0.5}$
Article or determiner errors	A	L8-NAIST$_{.30}$; tokens; naive	57.58	3.66	14.59	68.97	12.20	35.71
		L8-NAIST$_{.30}$; tokens; NP	54.44	9.44	**27.87**	58.33	20.83	42.89
		L8-NAIST$_{.30}$; tokens; NP$_{verb,prep}$	51.06	9.25	26.82	53.03	20.71	40.42
		L8-NAIST$_{.30}$; tokens; POS$_{5000}$	54.17	7.51	24.16	65.08	23.70	**48.24**
		L8-NAIST$_{.30}$; tokens; AWC$_{50}$	54.55	8.09	25.39	62.26	19.53	**43.31**
	B	L8-NAIST$_{.30}$; AWC$_{50}$; tokens	53.01	8.48	25.85	62.30	21.84	45.45
		L8-NAIST$_{.30}$; AWC$_{50}$; tokens+POS	45.22	10.02	26.56	60.20	32.42	**51.39**
		L8-NAIST$_{.30}$; AWC$_{50}$; tokens+POS+mix$_{POS}$	40.22	13.87	**29.15**	54.78	33.69	48.69
		L8-NAIST$_{.30}$; AWC$_{50}$; tokens+AWC	44.92	10.21	26.74	63.64	28.16	**50.83**
		L8-NAIST$_{.30}$; AWC$_{50}$; tokens+AWC+mix$_{AWC}$	42.31	12.72	**28.87**	56.38	29.44	47.66
		L8-NAIST$_{.30}$; AWC$_{50}$; tokens+POS+AWC	30.10	17.34	26.24	41.74	49.51	43.09
	C	L8-WEB$_{.30}$; POS$_{5000}$; tokens+POS	47.26	13.29	31.28	57.89	35.87	**51.56**
		L8-WEB$_{.30}$; AWC$_{50}$; tokens+AWC	42.13	14.45	30.46	55.56	33.52	49.10
Preposition errors	D	L8-WEB$_{.20}$; tokens	42.42	7.37	21.74	70.00	17.36	43.57
		L8-WEB$_{.20}$; tokens+POS	40.00	8.42	**22.86**	59.46	18.03	40.74
		L8-WEB$_{.20}$; tokens+POS+mix$_{POS}$	34.09	7.89	20.49	48.72	14.96	33.57
		L8-WEB$_{.20}$; tokens+AWC	36.36	8.42	21.86	67.65	19.66	**45.45**
		L8-WEB$_{.20}$; tokens+AWC+mix$_{AWC}$	24.49	6.32	15.54	50.94	21.95	40.30
		L8-WEB$_{.20}$; tokens+POS+AWC	37.21	8.42	22.10	68.75	19.13	45.27
	E	L8-WEB$_{.20}$; POS$_{5000}$; tokens+POS	34.78	8.42	21.39	58.14	20.66	42.66
		L8-WEB$_{.20}$; AWC$_{50}$; tokens+AWC	36.00	9.47	23.08	75.68	23.73	**52.63**

Table 3: Results on the CoNLL-2013 and CoNLL-2014 test sets for article and preposition error correction. Models are described by three attributes separated by semicolon: training data with specified error rate, method for article omission detection (except section D) and feature set. All models use *source* feature.

A naive method gives the lowest $F_{0.5}$ scores due to the high precision but low recall. Using a lower error rate in the training data shows a similar effect. Methods based on a shallow parser (NP) are more effective without augmenting them with spaces after each verb and preposition (NP$_{verb,prep}$) on both test sets. The proposed methods that compare surrounding context are significantly better on TS-2014 and reach slightly lower results on TS-2013. It is unclear why AWC n-grams are more effective than POS n-grams for TS-2013 and vice versa for TS-2014.

We also experiment with applying the proposed methods to handle missed preposition errors (Table 3, section E). This increases the recall, since it enables making corrections that can not be detected otherwise, but may reduce precision.

5.2. Different feature sets and final models

Next, we compare different feature sets (sections B and D in Table 3). For article models we chose a method of detecting omissions that uses AWC n-grams due to its speed and simplicity. For preposition models we used L8-WEB corpus as training data to get a sufficient number of training examples.

Models trained only on lexical features result in $F_{0.5}$ values that are slightly lower than results achieved by models that use more complex features. Using POS or AWC n-grams shows improvement in performance for both, article and preposition models. Although adding mixed n-grams is shown to improve performance in contextual spell checking, in our experiments it has a positive effect only for articles on TS-2013.

Training article models on the L8-WEB corpus (section C) shows further improvement since more training examples are used. It also shows that POS tags (51.56) are more effective in article error correction than AWC tags (49.10).

For preposition errors (section E), the highest result on TS-2014 (52.63) is achieved by a model using tokens and AWC tags and handling preposition omissions. Further investigation of automatic word classes and various numbers of classes is required.

5.3. Top systems from CoNLL-2014 shared task

The best system (Felice et al., 2014) (CAMB) participating in CoNLL-2014 shared task uses a hybrid approach, which includes both a rule-based and an SMT system augmented by a large web-based language model. The system of Rozovskaya et al. (2014) (CUUI) for article error correction makes use of the averaged perceptron algorithm and POS-tagger and chunker outputs to generate some of its features and correction candidates. For preposition errors a naive Bayes classifier is trained on n-grams counts from the Google n-gram corpus. AMU (Grundkiewicz and Junczys-Dowmunt, 2014) is a phrase-based SMT system combining large training resources, task-specific parameter tuning and features.

In addition to the NUCLE and test set from CoNLL-2013, all systems make use of other resources that are significant in size. The CAMB system uses Cambridge Learner Corpus. A module for preposition error correction in the CUUI system is trained on the Google 1T 5-gram Corpus. The AMU system is trained on data scraped from Lang-8 in similar size to our L8-WEB corpus.

System outputs submitted by participants contain corrections of errors of various types. Thus, we removed cor-

System	ArtOrDet			Prep		
	P	**R**	$\mathbf{M}^2_{0.5}$	**P**	**R**	$\mathbf{M}^2_{0.5}$
CAMB	39.00	65.00	**42.39**	41.15	51.63	**42.89**
CUUI	28.41	72.06	32.32	32.04	26.61	30.78
AMU	40.54	25.28	36.17	46.05	28.00	40.79
this work	57.89	35.87	**51.56**	75.68	23.73	**52.63**

Table 4: Top systems from CoNLL-2014 shared task.

rections that do not concern words from confusion sets (i.e. from system outputs we extracted corrections that concern article or preposition errors only), similarly as reported for the official test sets. Results are presented in Table 4.

Our best model for article error correction trained on token and POS features significantly beats the CAMB system by nearly 10% F-score (42.39 vs. 51.56). The top preposition model that uses AWC features and handles preposition omissions outperforms the top system from CoNLL-2014 in similar amount (42.89 vs. 52.63). We generally achieve a higher precision and lower recall than other systems.

6. Conclusions and future work

In this paper we reinvestigated the classifier-based approach in grammatical error correction by reducing the linguistic knowledge hidden in many aspects of system development. We have shown that state-of-the-art results can be achieved without applying a multitude of heuristic rules, complex feature engineering, and advanced NLP tools.

Although, for article error correction the best performance is achieved by models trained on POS n-grams, AWC n-grams also outperform lexical features and top systems participating in the CoNLL-2014 shared task. For preposition error correction, models that use AWC features and allow preposition insertions outperform other systems. Our results have shown that the proposed simple contextual method for detecting omitted articles is competitive with methods relying on chunker outputs.

This work allows to believe that automatic word classes trained with unsupervised methods are promising substitution for part-of-speech tags at least in some applications.

In the future, we plan a deeper examination of the application of automatic word classes to GEC. Other models for unsupervised learning of word representations should be tested, as well as different numbers of word clusters.

Acknowledgements

This work is funded by the National Science Centre, Poland (Grant No. 2014/15/N/ST6/02330).

7. References

Cahill, Aoife, Nitin Madnani, Joel R Tetreault, and Diane Napolitano. 2013. Robust systems for preposition error correction using Wikipedia revisions. In *NAACL-HLT*.

Dahlmeier, Daniel and Hwee Tou Ng. 2012. Better evaluation for grammatical error correction. In *NAACL-HLT*.

Dahlmeier, Daniel, Hwee Tou Ng, and Siew Mei Wu. 2013. Building a large annotated corpus of learner english: The NUS Corpus of Learner English. In *BEA8 Workshop*.

Fan, Rong-En, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. 2008. LIBLINEAR: A library for large linear classification. *JMLR*, 9:1871–1874.

Felice, Mariano, Zheng Yuan, Øistein E Andersen, Helen Yannakoudakis, and Ekaterina Kochmar. 2014. Grammatical error correction using hybrid systems and type filtering. *CoNLL*, pages 15–24.

Gamon, Michael, Jianfeng Gao, Chris Brockett, Alexandre Klementiev, William B Dolan, Dmitriy Belenko, and Lucy Vanderwende. 2008. Using contextual speller techniques and language modeling for ESL error correction. In *IJCNLP*, volume 8.

Grundkiewicz, Roman and Marcin Junczys-Dowmunt. 2014. The AMU system in the CoNLL-2014 shared task: Grammatical error correction by data-intensive and feature-rich statistical machine translation. *CoNLL*, pages 25–33.

Han, Na-Rae, Martin Chodorow, and Claudia Leacock. 2006. Detecting errors in english article usage by non-native speakers. *JNLE*, 12(02):115–129.

Han, Na-Rae, Joel R Tetreault, Soo-Hwa Lee, and Jin-Young Ha. 2010. Using an error-annotated learner corpus to develop an ESL/EFL error correction system. In *LREC*.

Leacock, Claudia, Martin Chodorow, Michael Gamon, and Joel Tetreault. 2010. Automated grammatical error detection for language learners. *Synthesis lectures on human language technologies*, 3(1):1–134.

Mikolov, Tomas, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*.

Mizumoto, Tomoya, Yuta Hayashibe, Mamoru Komachi, Masaaki Nagata, and Yu Matsumoto. 2012. The effect of learner corpus size in grammatical error correction of ESL writings. In *COLING*.

Ng, Hwee Tou, Siew Mei Wu, Yuanbin Wu, Christian Hadiwinoto, and Joel Tetreault. 2013. The CoNLL-2013 shared task on grammatical error correction. In *CoNLL*.

Ng, Hwee Tou, Siew Mei Wu, Ted Briscoe, Christian Hadiwinoto, Raymond Hendy Susanto, and Christopher Bryant. 2014. The CoNLL-2014 shared task on grammatical error correction. In *CoNLL*.

Rozovskaya, Alla and Dan Roth. 2010. Generating confusion sets for context-sensitive error correction. In *EMNLP*.

Rozovskaya, Alla, Kai-Wei Chang, Mark Sammons, and Dan Roth. 2013. The University of Illinois system in the CoNLL-2013 shared task. In *CoNLL*.

Rozovskaya, Alla, Kai-Wei Chang, Mark Sammons, Dan Roth, and Nizar Habash. 2014. The Illinois-Columbia system in the CoNLL-2014 shared task. pages 34–42.

Tetreault, Joel R and Martin Chodorow. 2008. The ups and downs of preposition error detection in ESL writing. In *COLING*.

Tetreault, Joel, Jennifer Foster, and Martin Chodorow. 2010. Using parse features for preposition selection and error detection. In *ACL*.